UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER OF PATENTS AND TRADEMARKS
Washington, D.C. 20231
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/282,907	03/31/1999	CHING-YUN CHAO	AT9-98-441	9256

7590 07/17/2002

WINSTEAD SECHREST & MINICK
100 CONGRESS AVENUE
SUITE 800
AUSTIN, TX 78701

EXAMINER
FLEURANTIN, JEAN B

ART UNIT	PAPER NUMBER
2172	13

DATE MAILED: 07/17/2002

Please find below and/or attached an Office communication concerning this application or proceeding.

UNITED STATES PATENT AND TRADEMARK OFFICE

COMMISSIONER FOR PATENTS
UNITED STATES PATENT AND TRADEMARK OFFICE
WASHINGTON, D.C. 20231
www.uspto.gov

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Paper No. 13

Application Number: 09/282,907
Filing Date: March 31, 1999
Appellant(s): CHAO ET AL.

MAILED

JUL 16 2002

Technology Center 2100

Ching-yun Chao
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed on April 18, 2002.

The appeal brief filed on April 18, 2002 has been entered.

(1) Real Party in Interest

A statement identifying the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

A statement identifying the related appeals and interferences which will directly affect or be directly affected by or have a bearing on the decision in the pending appeal is contained in the brief.

(3) Status of Claims

The statement of the status of the claims contained in the brief is correct.

(4) Status of Amendments After Final

No amendment after final has been filed.

(5) Summary of Invention

The summary of invention contained in the brief is correct.

(6) Issues

The appellant's statement of the issues in the brief is correct.

(7) Grouping of Claims

Appellant's brief includes a statement that claims do not stand or fall together and provides reasons as set forth in 37 CFR 1.192(c)(7) and (c)(8).

(8) Claims Appealed

The copy of the appealed claims contained in the Appendix to the brief is correct.

(9) Prior Art of Record

The following is a listing of the prior art of record relied upon in the rejection of claims under appeal.

✓5,956,489	San Andres et al.	9-1999

(10) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims 1, 9, 15 and 21:

Claims 1-25 are rejected under 35 U.S. C. 103 (a) as being unpatentable over San Andres et al. (US Pat. No. 5,956,489). This rejection is set forth in prior Office Action, mailed on September 25, 2001, Paper No. 7.

Claim Rejections - 35 U.S.C. § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claims 1-30 are rejected under 35 U.S. C. 103 (a) as being unpatentable over San Andres et al. (US Pat. No. 5,956,489).

As per claim 1, San Andres substantially teaches a method for maintaining a consistent set of replicas of a database within a computer cluster (thus, enabling each application server to maintain a replicated copy of service content data, which is readable as maintaining a consistent

set of replicas of a database within a computer cluster) (see abstract, lines 5-6) as claimed comprises the steps of each node in the computer cluster receiving a database update request (thus, once the content of the application server is brought up to date the application server is placed in a state which allows it to receive client request, which is readable as each node in the computer cluster receiving a database update request)(see col. 3, lines 8-11, 26-30);

detecting an out-of-sync condition as a result of a different functional outcome (thus, when different application servers of a service group process the same update transaction differently the arbiter resolves the conflict by determining the final outcome of the transaction for the service group as whole and taking by any application servers off line that are in conflict with this final outcome, which is readable as detecting an out-of-sync condition as a result of a different functional outcome) (see col. 2, lines 55-60). But, San Andres explicitly does not indicate the step of each node in the computer cluster voting based on a functional outcome of the database update request. However, San Andres suggest the step of the arbiter monitors the outcome of the transaction on each server by checking the status codes returned by the servers, when one server of the service group processes the dispatched transaction differently than the other servers the arbiter uses a voting scheme to decide which server or servers are to be taken off line service group, the arbiter uses a majority rules voting scheme under the majority rules scheme, if the majority number servers of the service group report a different outcome than others servers the majority servers are treated as being inconsistent with final outcome and taken off line; which is readable as each node in the computer cluster voting based on a functional outcome of the database update request (see col. 19, lines 43-55). It would have been obvious to a person of ordinary skill in the art at the time the invention was made to modify the teaching of San Andres with the step of each node in the computer cluster voting based on a functional outcome of the database update request. This modification would allow the teachings of San

Andres to provide access to identical data and so that the on line service appears the same to all end users (col. 1, lines 61-61).

As per claim 2, San Andres substantially teaches a method as claimed, wherein the out-of-sync condition is an error (thus, the arbiter uses a majority rules voting scheme under the majority rules scheme, if the majority number servers of the service group report a different outcome than others servers the majority servers are treated as being inconsistent with final outcome and taken off line; which is readable as wherein the out-of-sync condition is an error) (see col. 19, lines 18-55).

As per claim 3, San Andres substantially teaches a method as claimed, further comprises the step of: refreshing the database in response to the detecting step (thus, when an application server of a service group receives a client request that indicates a modification to replicated service content data the server generates an update transaction and sends the update transaction to the arbiter, which is readable as refreshing the database in response to the detecting) (see col. 3, lines 26-30).

As per claim 4, San Andres substantially teaches a method as claimed, further comprises the step of: resetting cluster membership in response to the detecting step (thus, when an application server of a service group receives a client request that indicates a modification to replicated service content data the server generates an update transaction and sends the update transaction to the arbiter, which is readable as resetting cluster membership in response to the detecting (see col. 3, lines 26-30).

As per claim 5, San Andres substantially teaches a method as claimed, further comprises the step of blocking further participation by the node having the out-of-sync condition in response to the detecting step (thus, when different application servers of a service group process the same update transaction differently the arbiter resolves the conflict by determining the final outcome of the transaction for the service group as whole and taking by any application servers

off line that are in conflict with this final outcome, which is readable as participation by the node having the out-of-sync condition in response to the detecting step (see col. 2, lines 55-60).

As per claim 6, San Andres substantially teaches a method as claimed, further comprises the step of: declaring an end-of-transaction state on update voting completion when the database update is being done in a transactional manner (thus, servers receive the update transaction from arbiter respond by processing the update transaction and returning a status code that indicates the success or failure, which is equivalent to declaring an end-of-transaction state on update voting completion when the database update is being done in a transactional manner) (see col. 19, lines 25-28).

As per claim 7, San Andres substantially teaches a method as claimed, further comprising the step of backing out an update when update voting does not meet a criteria established for success (thus, servers receive the update transaction from arbiter respond by processing the update transaction and returning a status code that indicates the success or failure, which is readable as backing out an update when update voting does not meet a criteria established for success) (see col. 19, lines 25-28).

As per claim 8, San Andres substantially teaches a method as claimed, wherein the criteria established for success is that no more than one node has inconsistent results (see col. 2, lines 55-64).

As per claim 9, in addition to the discussion in claim 1, San Andres teaches a method for maintaining a consistent set of replicas of a database within a computer cluster (see col. 1, lines 60-61) as claimed, comprises the steps of: broadcasting an update to a database shared among a plurality of nodes in the computer cluster (thus, the arbiter records the update transaction in a service group specific transaction log and forwards the transaction for immediate processing to every application server in the group, which is equivalent to broadcasting an update to a database shared among a plurality of nodes in the computer cluster) (see col. 3, lines 30-34);

applying the update to a local copy of the database at each of the plurality of nodes in the computer cluster (thus, all duplicated servers of the service group maintain local copies of the service's content data and provide user access to such data, which is readable as applying the update to a local copy of the database at each of the plurality of nodes in the computer cluster) (see col. 9, lines 19-21);

comparing, by all of the other nodes in the computer cluster, the update results to results of application of the update to the local copy of the database (thus, a conflict resolution feature for resolving transaction processing conflicts between application servers, when different application servers of a service group process the same update transaction differently the arbiter resolves the conflict by determining the final outcome of the transaction for the service group as whole and taking by any application servers off line that are in conflict with this final outcome, which is readable as comparing, by all of the other nodes in the computer cluster, the update results to results of application of the update to the local copy of the database) (see col. 2, lines 53-60). But, San Andres explicitly does not indicate the step of the voting, by all of the other nodes in the computer cluster to approve update if a match results from the comparison. However, San Andres shows implicitly the step of the whenever the arbiter replicates a transaction the arbiter monitors the outcome of the transaction on each server of the service group to ensure consistent processing of the transaction by all such servers, when one or more servers indicates a different outcome than the other servers of the service group the arbiter uses a voting scheme to resolve the conflict between the servers a preferred voting scheme is described below under the heading status codes and conflict resolution; which is read as voting, by all of the other nodes in the computer cluster to approve update if a match results from the comparison (see col. 17, lines 10-19). Thus, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to modify the teaching of San Andres with the step of voting, by all of the other nodes in the computer cluster to approve update if a match results from

the comparison. This modification would allow the teachings of San Andres to improve the accuracy and the reliability of the error detection protocol.

As per claims 10 and 16, San Andres substantially teaches a method as claimed, further comprising the step of: voting, by any one of the other nodes in the computer cluster, to continue with update process if a match does not result from the comparison (thus, when inconsistent transaction results are reported by different application servers the transaction replication service uses a voting scheme to decide which application servers are to be deemed 'consistent'; which is readable as voting, by any one of the other nodes in the computer cluster, to continue with update process if a match does not result from the comparison) (see abstract, lines 10-14).

As per claims 11, 17, and 22 San Andres substantially teaches a method as claimed, further comprising the step of broadcasting an approval of the update to the database if all of the other nodes vote to approve the update (thus, when a new application server is brought on line previously dispatched update transactions stored in the transaction log are dispatched in sequence to the new server to bring the new server's content data up to date, which is readable as broadcasting an approval of the update to the database if all of the other nodes vote to approve the update) (see abstract, lines 16-20).

As per claims 12, 18, and 23 San Andres substantially teaches a method as claimed, further comprises the step of if more than one of the plurality of nodes votes to continue, performing a recovery process (see abstract, lines 11-13).

As per claims 13, 19, and 24 San Andres substantially teaches a method as claimed, wherein the recovery process further comprises the step of if more than a specified number of the nodes voted to continue, backing out the update to the database (see col. 20, lines 26-42).

As per claims 14 and 20, San Andres substantially teaches a method as claimed 14. The method as recited in claim 12, wherein the recovery process further comprises the step of: if less

than a specified number of the nodes voted to continue, performing the recovery process on the specified number of the nodes (see col. 17, lines 10-19).

As per claim 15, in addition to the discussion in claims 1 and 9, San Andres substantially teaches a computer cluster operable for maintaining a consistent set of replicas of a database within the computer cluster as claimed, comprises: a group services client operable for broadcasting an update to a database shared among a plurality of nodes in the computer cluster (thus, architecture of on line services network in which a preferred embodiment of a transaction replication system and method are employed, which is readable as a group services client operable for broadcasting an update to a database shared among a plurality of nodes in the computer cluster) (see figure 1, col. 3, lines 50-53);

the plurality of nodes coupled to the computer cluster operable for applying the update to a local copy of the database at each of the plurality of nodes in the computer cluster (thus, a conflict resolution feature for resolving transaction processing conflicts between application servers, when different application servers of a service group process the same update transaction differently the arbiter resolves the conflict by determining the final outcome of the transaction for the service group as whole and taking by any application servers off line that are in conflict with this final outcome, which is readable as the plurality of nodes coupled to the computer cluster operable for applying the update to a local copy of the database at each of the plurality of nodes in the computer cluster) (see col. 2, lines 53-60).

As per claim 21, in addition to the discussion in claim 9, San Andres substantially teaches a computer program product adaptable for storage on a computer readable medium, the computer program product operable for maintaining a consistent set of replicas of a database within a computer cluster as claimed, comprises the program steps of voting, by any one if the other nodes in the computer cluster, to continue with update process if a match does not result from the comparison (the transaction replication service uses a voting scheme to decide which application

servers are to be deemed 'consistent'; which is readable as voting, by any one if the other nodes in the computer cluster, to continue with update process if a match does not result from the comparison) (see abstract, lines 11-13).

As per claim 25, San Andres substantially teaches a computer program product as claimed, wherein the recovery process further comprises the program step of if less than a specified number of the nodes voted to continue, performing the recovery process on the specified number of the nodes (see col. 17, lines 10-19).

(11) Response to Arguments

1. As per claims 1, 9, 15, and 21 Applicant argues that the San Andres reference does not teach or suggest:

In response to applicant's argument on pages 3-6, that there is no suggestion to combine the references, the examiner recognizes that obviousness can only be established by combining or modifying the teachings of the prior art to produce the claimed invention where there is some teaching, suggestion, or motivation to do so found either in the references themselves or in the knowledge generally available to one of ordinary skill in the art. See *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988) and *In re Jones*, 958 F.2d 347, 21 USPQ2d 1941 (Fed. Cir. 1992). In this case, San Andres provides user access to service content data that is updated by the on line service on a transaction by transaction basis, and permits users to read and download messages for review by other users; see col. 1, lines 38-43.

Also, San Andres further teaches steps of what is also needed is an efficient mechanism for bringing the content of an application server up to date with that of other application servers sot that new application servers can be added to service groups, and so that existing application servers can efficiently be taken off line for maintenance; see col. 2, lines 24-30.

In response to applicant's argument on pages 7-9, that the examiner's conclusion of obviousness is based upon improper hindsight reasoning, it must be recognized that any

judgment on obviousness is in a sense necessarily a reconstruction based upon hindsight reasoning. But so long as it takes into account only knowledge which was within the level of ordinary skill at the time the claimed invention was made, and does not include knowledge gleaned only from the applicant's disclosure, such a reconstruction is proper. See *In re McLaughlin*, 443 F.2d 1392, 170 USPQ 209 (CCPA 1971).

Applicant stated on page 9, that San Andres does not teach 'detecting an out-of-sync condition as a result of a different functional outcome'. However, Examiner disagrees because San Andres includes steps of each time an update transaction is dispatched by the arbiter the arbiter monitors the outcome 'success or failure' of the transaction on each server by checking the status codes returned by the server, which is readable as detecting an out-of-sync condition as a result of a different functional outcome, see col. 19, lines 43-46. Thus, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to modify the teaching of San Andres with the step of each node in the computer cluster voting based on a functional outcome of the database update request. This modification would allow the teachings of San Andres to improve the accuracy of the error detection protocol.

Applicant stated on pages 11-12, that San Andres does not teach 'refreshing the database in response to the detecting step'. However, Examiner disagrees because San Andres includes steps of when an application server of a service group receives a client request that indicates a modification to replicated service content data the server generates an update transaction and sends the update transaction to the arbiter, which is readable as refreshing the database in response to the detecting; see col. 3, lines 26-30.

Applicant stated on pages 10 and 11, that San Andres does not teach 'the out-of-sync condition is an error'. However, Examiner disagrees because San Andres includes steps of the arbiter uses a majority rules voting scheme under the majority rules scheme, if the majority number servers of the service group report a different outcome than others servers the majority

servers are treated as being inconsistent with final outcome and taken off line; which is readable as the out-of-sync condition is an error; see col. 19, lines 18-55.

Applicant stated on pages 13 and 14, that San Andres does not teach 'voting by all of the other nodes in the computer cluster, to approve update if a match results from the comparison'. However, Examiner disagrees because San Andres includes the steps of the Arbiter monitors the outcome of the transaction on each server 120 of the service group to ensure consistent processing of the transaction by all such servers, which is readable as voting by all of the other nodes in the computer cluster; see col. 17, lines 11-14.

Applicant stated on pages 14, 17, 18 and 21, that 'voting, by all of the other nodes in the computer cluster, to approve update if a match results from the comparison'. However, Examiner disagrees because San Andres includes the step the whenever the arbiter replicates a transaction the arbiter monitors the outcome of the transaction on each server of the service group to ensure consistent processing of the transaction by all such servers, when one or more servers indicates a different outcome than the other servers of the service group the arbiter uses a voting scheme to resolve the conflict between the servers a preferred voting scheme is described below under the heading status codes and conflict resolution; which is read as voting, by all of the other nodes in the computer cluster, to approve update if a match results from the comparison; see col. 17, lines 10-19.

Applicant stated on pages 15 and 16, that San Andres does not teach or suggest 'applying the update to a local copy of the database at each of the plurality of nodes in the computer cluster'. However, Examiner disagrees because San Andres includes the steps of services can use this feature to perform update to replicated copies of service content data different online services can advantageously use the arbiter's transaction replica feature for different purposes; which is readable as applying the update to a local copy of the database at each of the plurality of nodes in the computer cluster; see col. 2, lines 44-47.

Applicant stated on page 16, that San Andres does not teach or suggest 'comparing, by all of the other nodes in the computer cluster, the update results to results of application of the update to the local copy of the database'. However, Examiner disagrees because San Andres includes the steps of the arbiter replicates service content data by dispatching atomic transactions 'which are generated by the arbitered services' to groups servers 120 these transactions are in the form of update commands (referred to herein as "update transactions") which when interpreted by the receiving service applications typically specify an update to a specific data entity or set of data entities each server 120 which receives the update transaction from the arbiter processes the transaction, which normally involves updating the server's locally-stored service content data; which is readable as comparing, by all of the other nodes in the computer cluster, the update results to results of application of the update to the local copy of the database; see col. 16, lines 50. Also, San Andres further teaches the steps of services can use this feature to perform update to replicated copies of service content data different online services can advantageously use the arbiter's transaction replica feature for different purposes; see col. 2, lines 44-47.

Applicant stated on page 19, that San Andres does not teach or suggest 'broadcasting an approval of the update to the database if all of the other nodes vote to approve the update'. However, Examiner disagrees because San Andres includes the steps of when a new application server is brought on line previously dispatched update transactions stored in the transaction log are dispatched in sequence to the new server to bring the new server's content data up to date, which is readable as broadcasting an approval of the update to the database if all of the other nodes vote to approve the update; see abstract, lines 16-20.

Examiner is entitled to give claim limitations their broadest reasonable interpretation in light of the specification.

Interpretation of Claims-Broadest Reasonable Interpretation

During patent examination, the pending claims must be 'given the broadest reasonable interpretation consistent with the specification.' Applicant always has the opportunity to amend the claims during prosecussion and broad interpretation by the examiner reduces the possibility that the claim, once issued, will be interpreted more broadly than is justified. In re Prater, 162 USPQ 541,550-51 (CCPA 1969).

Respectfully submitted,

Jean Bolte Fleurantin

JBF/

Jean M. Corrielus

HOSAIN T. ALAM
PRIMARY EXAMINER